



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/670,898	09/25/2003	William J. Masek	LOT920030024/US1	5987
45544 7590 05/29/2010 HOFFMAN WARNICK LLC 75 STATE ST 14TH FLOOR ALBANY, NY 12207				
EXAMINER MITCHELL, JASON D				
ART UNIT 2193		PAPER NUMBER		
NOTIFICATION DATE 05/20/2010		DELIVERY MODE ELECTRONIC		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

PTOCommunications@hoffmanwarnick.com

Office Action Summary

Application No.

10/670,898

Applicant(s)

MASEK ET AL.

Examiner

Jason Mitchell

Art Unit

2193

Period for Reply -- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 08 March 2010.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-9, 11-18 and 20-26 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-9, 11-18 and 20-26 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB-06)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

This action is in response to a request for continued examination filed on 3/8/10.

Claims 1-9, 11-18 and 20-26 are pending in this application.

Response to Arguments

Applicant's arguments filed 3/8/10 have been fully considered but they are not persuasive.

Rejection of Claims 1-5, 7-9, 11-14, 16-18, 20-23 and 25-26

In the first par. on pg. 8, the applicants state:

...with respect to independent claim 1, Applicants assert that the cited references fail to teach or suggest, inter alia, "... each of the plurality of instances of the test application run within a single process, sharing all services and memory space with others of the plurality of instances, without requiring multiple processes to instantiate the plurality of instances within". The Office equates the process of the claimed invention with the Visual Basic Form of Duggan that contains multiple instances of a custom control for allocating a resource. However, the all of the instances in Duggan's Visual Basic Form do not share all services and memory with each other. Rather, one instance of the Visual Basic form is used for each session. Col. 22, lines 45-63. As such, Duggan requires multiple sessions each requiring a separate instance of a custom control from a form. Thus, Duggan fails to teach or suggest that its concurrent sessions run without requiring multiple processes to instantiate the plurality of instances within. Claims 9 and 18 include similar features. Firth fails to cure this deficiency. Accordingly, Applicants respectfully request that the Office withdraw its rejection.

The examiner respectfully disagrees. First, it is noted that the previous rejection did not map Duggan's "Visual Basic Form" to the claimed "process" as asserted by the applicants. Instead the claimed process was rejected over the process in which Duggan's "basic module 12" (see e.g. col. 21, lines 52-61) is executed. The "Visual

Basic Form" described in Duggan's col. 22, lines 45-62 establishes connections between the individual threads and the server being tested (e.g. "One instance of the custom control is used to control each client connection"). Further, this communication strategy appears to be similar to the one disclosed, but not claimed, by the applicants (see e.g. the applicants' par. [0023] "Each thread in turn is associated with a different connection to server 14"; also see par. [0018] "connectivity could be provided by conventional TCP/IP sockets-based protocol"). Regardless, the point is largely moot because in an effort to further the prosecution, new grounds of rejection have been presented in this action.

In the par. bridging pp. 8 and 9 the applicants state:

With further respect to independent claim 1, Applicants continue to respectfully assert, in addition to the above arguments, that the cited references also fail to teach or suggest, *inter alia*, "... identifying application protocol interfaces (APIs) ..., prior to the instantiating step... [and] providing a test script capable of invoking the APIs ...". Claim 1. The Office admits that Duggan does not explicitly disclose that its command module is implemented as APIs. Rather, the Office cites a passage of Firth that teaches, generically, that APIs exist, reciting "functions in the Internet API reside in a dynamic link library (DLL)." Col. 2, lines 63-67. To this extent, the Office attempts to replace whole pages of Duggan that describe the formation of scripts with one generic sentence describing where an API is stored. Applicants respectfully submit that the reference to an API in Firth has nothing to do with creating testing of application programs, as in Duggan, and any attempt to incorporate this generic API into the Duggan Visual Basic GUI based system of Duggan would, at best, lead to undue experimentation and yield unpredictable results. Accordingly, Applicants request that the Office withdraw the rejection of claim 1.

The examiner respectfully disagrees. Duggan discloses providing a set of functions to the test scripts (col. 13, lines 43-46 "the command module 14 contains a number of different commands, each of which performs a different user function of the

application program under test"). APIs are a basic coding technique, well known in the software development arts and commonly used to provide a set of functions for use by other pieces of code (see e.g. Firth, col. 2, lines 49-52 "An ... application program interface (API) ... including a set of reentrant ... functions is used ... by multiple application programs"). Accordingly, an API is a well known and commonly used method of providing the functionality described by Duggan. Further, contrary to the applicants assertion, such a modification does not "replace whole pages of Duggan". Instead modifying Duggan to take advantage of well known APIs only replaces a single sentence describing a single embodiment (i.e. col. 14, lines 22-23 "the command module is implemented as a Visual Basic 5.0 code module"). In other words, use of an API only changes the way in which the code is written and packaged and does not represent a change to the user level functionality provided to a tester.

The arguments regarding claims 2-9, 11-18 and 20-26 are similarly unpersuasive.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 1-5, 7-9, 11-14, 16-18, 20-23 and 25-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 6,002,871 to Duggan et al. (Duggan) in view of US

2004/0199825 A1 to Dinker et al. (Dinker) in view of US 7,062,755 B2 to Partamian et al. (Partamian) in view of US 5,987,517 to Firth et al. (Firth).

Regarding Claims 1, 9 and 18: Duggan discloses:

providing a test application that satisfies reentrancy requirements (*col. 21, lines 57-61 'Each session is ... reentrant'*) on a client (*col. 5, lines 18-21 'the test tool ... runs on a single computer'*);

identifying command modules associated with the test application (*col. 12, lines 21-23 "A list box 272 contains a list of all of the commands in the command module created for testing a given application program", the command module is inherently identified to the list box in order for the list box to present all of the commands from that module; col. 14, lines 22-28 "the command module is implemented as a Visual Basic 5.0 code module, Each command of the command module comprises a Visual Basic subroutine that contains the instructions for the execution segment of the command"*);

providing a test script capable of invoking the command modules (*col. 13, lines 59-62 "a test operator [can] create test scripts containing ... command module commands"*); and

instantiating, via the test script, a plurality of instances of the test application using threads (*col. 21, lines 57-61 'Each session is executed as a separate thread'; col. 21, lines 46-50 "handles the execution of a test run based on a ... test script"*).

Duggan discloses instantiating and executing a plurality of instances of the test application under the control of a single application (*col. 21, lines 53-61* “*The basic module 12 is also responsible for initiating multiple, concurrent sessions ... Each session is executed as a separate thread ... It is the multi-threaded, reentrant nature of the test tool program code*”). However, Duggan does not explicitly disclose the instantiating and execution of each of the plurality of instances of the test application occur within a single process, without requiring multiple processes to instantiate the plurality of instances within.

Dinker teaches a testing program which instantiates and executes each of a plurality of instances of the test application as one of a plurality of threads in a process (*par. [0028]* “Each test agent 110 may be implemented as a multithreaded application”; *par. [0031]* “multi-threaded test processes (i.e., test agents 110)”).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to instantiate and execute each of the plurality of instances of the test application (Duggan *col. 21, lines 53-61* “*initiating multiple, concurrent sessions ... Each session is executed as a separate thread*”; Dinker *par. [0028]* “*Each test agent 110 may be implemented as a multithreaded application*”; *par. [0031]* “*multi-threaded test processes (i.e., test agents 110)*”) within a single process without requiring multiple processes to instantiate the plurality of instances within (e.g. by only implementing Duggan's single test agent 110 instead of Dinker's multiple test agents). Those of

ordinary skill in the art would have been motivated to do so as one of a finite set of known and implementable methods of providing the disclosed functionality (*i.e. the threads are either implemented in the same process or different processes*) which would produce only the expected results (Duggan col. 21, lines 53-61 *'initiating multiple, concurrent sessions ... Each session is executed as a separate thread ... It is the multi-threaded, reentrant nature of the test tool program code'; par. [0031] "multi-threaded test processes (i.e., test agents 110)"*).

Duggen discloses the code of the individual threads can safely share services and memory space with the other threads (col. 21, lines 53-61 *"each session is reentrant"*). However, Duggen and Dinker do not explicitly teach sharing all services and memory space among the plurality of instances.

Partamian teaches sharing all services and memory space among the plurality of instances (col. 3, lines 12-17 *"A process may have one or a plurality of threads. Threads in the same process share information using memory, atomic instructions, mutexes, semaphores, etc."*).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to share all services and memory space (Partamian col. 3, lines 12-17 *"Threads in the same process share information using memory, atomic instructions, mutexes, semaphores, etc."*) among the plurality of threads (Duggen col. 21, lines 53-61

'initiating multiple, concurrent sessions ... Each session is executed as a separate thread ... each session is reentrant'; par. [0031] "multi-threaded test processes (i.e., test agents 110)". Those of ordinary skill in the art would have been motivated to do so as one of a finite set of known and implementable method of providing the disclosed functionality (i.e. either the threads share memory and services or they don't) which would have resulted in only the expected results (Duggan col. 21, lines 53-61 'initiating multiple, concurrent sessions ... Each session is executed as a separate thread ... each session is reentrant'; Partamian col. 3, lines 12-17 "Threads in the same process share information using memory, atomic instructions, mutexes, semaphores, etc.")).

Duggan, Dinker and Partamian do not explicitly teach the command module implemented as APIs.

Firth teaches the use of APIs (col. 2, lines 63-67 *"functions in the Internet API reside in a dynamic link library (DLL)"*).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to implement Duggan's command module (col. 14, lines 22-28 *"the command module is implemented as a Visual Basic 5.0 code module"*) as an API and to provide a data entry field in the GUI to identify particular API's for use with the application under test. Those of ordinary skill in the art would have been motivated to do so because Firth's APIs "eliminate the need to embed source code directly in an application

program to manage Internet application protocols" (col. 2, lines 64-67; also see Duggan col. 16, lines 9-15 "each command simulates a real user's interaction ... by generating ... an HTTP request") and thus provide further abstraction for Duggan's test script development (see e.g. col. 13, lines 59-67 "No knowledge of the underlying programmed instruction of the command module is needed by a test operator"; col. 14, lines 2-4 "The command module is rewritten and/or customized for each different application program to be tested").

Regarding Claim 2: The rejection of claim 1 is incorporated; further Duggan discloses; and

upon execution, the test script instantiates the plurality of instances of the test application (col. 5, line 67-col. 6, line 3 'the test tool program executes multiple concurrent sessions') using threads (col. 21, lines 57-61 'Each session is executed as a separate thread') within the single process (col. 21, lines 53-57 'The basic module 12 is also responsible for initiating multiple, concurrent sessions'; col. 21, lines 57 "It is the multi-threaded, reentrant nature of the test tool program code").

Regarding Claims 3, 14 and 23: The rejection of claims 1, 9 and 18 are incorporated respectively, further; Duggan discloses the server application is a network application (col. 5, lines 9-12 'a test tool for testing application programs ... over a network').

Regarding Claims 4, 12 and 21: The rejection of claims 1, 9 and 18 are incorporated respectively, further; Duggan discloses the reentrancy requirements dictates that the plurality of instances of the test application be run within the single process without interfering with each other (*col. 21, lines 57-61 'reentrant nature of the test tool'*).

Regarding Claims 5, 13 and 22: The rejection of claims 1, 9 and 18 are incorporated respectively, further; Duggan discloses each of the plurality of instances of the test application corresponds to a separate thread (*col. 21, lines 57-61 'Each session is executed as a separate thread'*), and wherein each of the separate threads is associated with a different connection to the server (*col. 5, line 66-col. 6, line 3 'A "session" refers to the execution of one test script, on one client connection'*).

Regarding Claims 7, 16 and 25: The rejection of claims 1, 9 and 18 are incorporated respectively, further; Duggan discloses the plurality of instances of the test application simulate use of the server application by a plurality of users (*col. 6, lines 47-51 'the test tool program ... is capable of executing test scripts ... based on a user list'*).

Regarding Claims 8, 17 and 26: The method of claim 1, 9 and 18 further comprising collecting data corresponding to the test (*col. 8, lines 4-6 'The test tool program ... provides four options for logging information'*).

Regarding Claims 11 and 20: The rejection of claims 9, and 18 are incorporated respectively, further; Duggan discloses, and wherein upon execution, the test script instantiates the plurality of instances of the test application (*col. 5, line 67-col. 6, line 3 'the test tool program executes multiple concurrent sessions'*) using threads (*col. 21, lines 57-61 'Each session is executed as a separate thread'*) within the single process (*col. 21, lines 53-57 'The basic module 12 is also responsible for initiating multiple, concurrent sessions'*).

Claims 6, 15 and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 6,002,871 to Duggan et al. (Duggan) in view of US 2004/0199825 A1 to Dinker et al. (Dinker) in view of US 7,062,755 B2 to Partamian et al. (Partamian) in view of US 5,987,517 to Firth et al. (Firth) in view of "The Java[™] Virtual Machine Specification" by Lindholm et al (Lindholm).

Regarding Claims 6, 15 and 24: The rejection of claims 1, 9 and 18 are incorporated respectively, further; Duggan does not disclose the process comprises a JAVA virtual machine.

Lindholm teaches that JAVA programs, which run on a JAVA virtual machine were well known at the time of the invention, and that JAVA programs and the JVM provided benefits known to those of ordinary skill in the art.

Art Unit: 2193

It would have been obvious to a person of ordinary skill in the art at the time of the invention to implement Duggan's 'test tool' and 'basic module' in the JAVA programming language and execute them on a JVM.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason Mitchell whose telephone number is (571)272-3728. The examiner can normally be reached on Monday-Thursday and alternate Fridays 7:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Bullock Lewis can be reached on (571) 272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Jason Mitchell/
Primary Examiner, Art Unit 2193